

## Pengenalan C++

C++ adalah bahasa lintas platform yang dapat digunakan untuk membuat aplikasi berkinerja tinggi.

C++ dikembangkan oleh Bjarne Stroustrup, sebagai perpanjangan dari bahasa [C](#).

C++ memberi pemrogram kontrol tingkat tinggi atas sumber daya dan memori sistem.

Bahasa ini diperbarui 4 kali utama pada tahun 2011, 2014, 2017, dan 2020 ke C++11, C++14, C++17, C++20.

## Mengapa Menggunakan C++

C++ adalah salah satu bahasa pemrograman paling populer di dunia.

C++ dapat ditemukan di sistem operasi saat ini, Antarmuka Pengguna Grafis, dan sistem tertanam.

C++ adalah bahasa pemrograman berorientasi objek yang memberikan struktur yang jelas untuk program dan memungkinkan kode untuk digunakan kembali, menurunkan biaya pengembangan.

C++ bersifat portabel dan dapat digunakan untuk mengembangkan aplikasi yang dapat disesuaikan dengan berbagai platform.

C++ menyenangkan dan mudah dipelajari!

Karena C++ dekat dengan [C#](#) dan [Java](#) , memudahkan programmer untuk beralih ke C++ atau sebaliknya.

Perbedaan antara C dan C++

C++ dikembangkan sebagai perpanjangan dari [C](#) , dan kedua bahasa memiliki sintaks yang hampir sama.

Perbedaan utama antara C dan C++ adalah bahwa C++ mendukung kelas dan objek, sedangkan C tidak.

# Pengertian Bahasa Pemrograman C++

**Bahasa Pemrograman C++** adalah bahasa pemrograman komputer yang bisa dipakai untuk membuat berbagai aplikasi (*general-purpose programming language*), dan merupakan pengembangan dari [bahasa pemrograman C](#).

Bahasa C++ kadang disebut juga sebagai "**C with Classes**". Ini karena perbedaan paling utama antara bahasa C dan C++ ada di dukungan ke pemrograman berbasis object (*object oriented programming*). Bahasa C++ sudah mendukung pemrograman berbasis object, sedangkan bahasa C belum.

Karena termasuk *general-purpose programming language*, yakni bahasa pemrograman yang bisa membuat berbagai aplikasi, bahasa pemrograman C++ bisa dipakai untuk membuat aplikasi desktop seperti antivirus, software pengolah gambar (*image processing*), aplikasi pengolah kata (*word processing*), hingga untuk membuat compiler bahasa pemrograman lain.

## Sejarah Singkat C++

Bahasa pemrograman C++ dirintis sejak tahun 1979 oleh **Bjarne Stroustrup**, seorang ilmuwan komputer asal **Denmark**. Pengembangan bahasa C++ terinspirasi dari penelitian tesis Phd yang dulunya pernah dikerjakan Stroustrup. Pada saat itu, **Stroustrup** tertarik dengan bahasa **Simula** yang sangat mudah dipakai untuk pembuatan aplikasi skala besar. Varian bahasa Simula, yakni Simula 67 dianggap sebagai bahasa pemrograman pertama yang menerapkan konsep pemrograman berbasis object.

Akan tetapi bahasa Simula dianggap terlalu lambat untuk penggunaan sehari-hari. Di sisi lain ada juga bahasa **BCPL** yang sangat cepat tapi kurang cocok untuk pembuatan aplikasi besar.

Ketika bekerja di laboratorium perusahaan **AT&T Bell Labs**, Bjarne Stroustrup memiliki masalah di pengembangan sistem operasi **UNIX**. Pada saat itu ia teringat tesis yang dulu dikerjakan dan ingin menerapkannya ke dalam bahasa C. Pemilihan bahasa C karena sudah cukup populer dan juga di eksekusi dengan cepat. Apalagi sistem operasi UNIX juga dibuat menggunakan bahasa C.

Akhirnya pada tahun 1982 Bjarne Stroustrup mulai menambah fitur-fitur baru ke dalam bahasa C, terutama dukungan *class* agar bahasa C bisa menggunakan paradigma *object oriented programming* (pemrograman berbasis object).

Stroustrup terinspirasi dari bahasa Simula dan juga bahasa pemrograman lain seperti **ALGOL 68**, **Ada**, **CLU** and **ML**.

Bahasa baru ini disebut sebagai **C++** yang diambil dari operator *increment* "++".

Operator increment biasa di pakai dalam bahasa pemrograman untuk menaikkan nilai variabel sebanyak 1 angka. Dengan demikian C++ juga berarti sebuah penambahan dari bahasa C standar.

Pada tahun 1985, Bjarne Stroustrup membuat buku yang sangat terkenal berjudul **C++ Programming Language**. Buku ini menjadi referensi utama dan menjadi “standar tidak resmi” dari bahasa C++ pada saat itu. Pada tahun yang sama, dikembangkan pula compiler komersil C++ pertama.

Seiring perkembangan waktu, organisasi standar ISO secara resmi mengeluarkan standar **C++98** pada tahun 1998, yang kemudian diikuti standar **C++03** di tahun 2003.

Setelah itu secara bergantian di rilis standar **C++11** di tahun 2011, standar **C++14** di tahun 2014 serta standar **C++17** di tahun 2017. Di akhir tahun 2020 ini di rencanakan rilis standar **C++ 20**.

Untuk penggunaan dasar, perbedaan standar C++ ini tidak akan begitu terasa. Seharusnya tutorial belajar bahasa C++ di Duniaikom ini tetap bisa dipakai pada semua compiler C++.

## Untuk Apa Belajar C++?

Saya yakin sebagian besar pembaca tutorial ini belajar bahasa C++ karena diwajibkan oleh guru atau dosen. Ini tidak mengherankan karena bahasa C++ termasuk bahasa dasar yang sering dipakai untuk mempelajari dasar algoritma dan pemrograman. Alternatif lain bisa juga menggunakan [bahasa Pascal](#) dan juga [bahasa C](#) (tergantung dosen senangnya pakai bahasa apa).

Pemilihan bahasa C++ sebagai bahasa pertama atau bahasa dasar sebenarnya juga sangat bermanfaat meskipun tidak diwajibkan dosen atau bagi yang belajar otodidak. Ini karena bahasa C dan C++ menjadi dasar dari banyak bahasa pemrograman modern.

Jika anda sudah familiar dengan *syntax* / aturan penulisan bahasa C++, akan sangat mudah beralih ke bahasa yang lebih modern seperti **Java**, **C#**, **PHP**, atau **JavaScript**. Karena bahasa-bahasa ini semuanya turunan dari bahasa C dan C++.

Di luar fungsinya untuk belajar pemrograman dasar, bahasa C++ juga bisa dipakai untuk membuat berbagai aplikasi. Yang paling menonjol menurut saya ada di **game programming**, yakni bahasa pemrograman yang dipakai untuk membuat game.

Game modern sudah terbilang sangat kompleks, apalagi untuk game desktop. Umumnya programmer game butuh sebuah **Game Engine** yang berisi kumpulan kode program siap pakai (disebut sebagai *library*). Fungsinya untuk membantu pembuatan game dengan cepat, tidak lagi membuat semuanya dari nol.

Game engine yang paling terkenal diantaranya **Unreal Engine**, **Unity**, **CryEngine**, dan **Construct**.

Diluar itu, C++ juga cocok dipakai untuk membuat aplikasi yang perlu banyak mengakses hardware seperti di perangkat **IoT** (*Internet of Things*), *embedded system*, hingga membuat aplikasi desktop berbasis grafis, terutama saat digabung dengan **Microsoft Visual Studio**.

## Sintaks C++

Mari kita pisahkan kode berikut untuk lebih memahaminya:

### Contoh

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello World!";
    return 0;
}
```

### Contoh dijelaskan

**Baris 1:** `#include <iostream>` adalah **pustaka file header** yang memungkinkan kita bekerja dengan objek input dan output, seperti `cout` (digunakan pada baris 5). File header menambahkan fungsionalitas ke program C++.

**Baris 2:** `using namespace std` berarti kita dapat menggunakan nama untuk objek dan variabel dari pustaka standar.

Jangan khawatir jika Anda tidak mengerti cara `#include <iostream>` dan cara `using namespace std` kerjanya. Anggap saja sebagai sesuatu yang (hampir) selalu muncul di program Anda.

**Baris 3:** Baris kosong. C++ mengabaikan ruang putih. Tapi kami menggunakannya untuk membuat kode lebih mudah dibaca.

**Baris 4:** Hal lain yang selalu muncul dalam program C++, adalah `int main()`. Ini disebut **fungsi**. Kode apa pun di dalam tanda kurung kurawalnya `{}` akan dieksekusi.

**Baris 5:** `cout` (diucapkan "see-out") adalah **objek** yang digunakan bersama dengan *operator penyisipan* ( `<<` ) untuk menampilkan/mencetak teks. Dalam contoh kita ini akan menampilkan "Hello World".

**Catatan:** Setiap pernyataan C++ diakhiri dengan titik koma ;.

**Catatan:** Tubuh `int main()` juga dapat ditulis sebagai:

```
int main () { cout << "Hello World! "; return 0; }
```

**Ingat:** Kompiler mengabaikan spasi putih. Namun, beberapa baris membuat kode lebih mudah dibaca.

**Baris 6:** `return 0` mengakhiri fungsi utama.

**Baris 7:** Jangan lupa untuk menambahkan kurung kurawal penutup `}` untuk benar-benar mengakhiri fungsi utama.

## Menghilangkan Namespace

Anda mungkin melihat beberapa program C++ yang berjalan tanpa pustaka namespace standar. Baris `using namespace std` dapat dihilangkan dan diganti dengan `std` kata kunci, diikuti oleh `::` operator untuk beberapa objek:

### Contoh

```
#include <iostream>
```

```
int main() {  
    std::cout << "Hello World!";  
    return 0;  
}
```